

caBIG Workspace Developer Project Form

Developers, please complete this form in advance of the caBIG kickoff meeting and return by e-mail to adamsm@mail.nih.gov. Completed forms will be made available to all participants in advance of the meeting to enhance workspace discussions. During our conversations with you, we expressed the aspect of your program that we would like you to develop in the first year of the caBIG pilot; it is this we are asking you to address - here and in your presentation.

Sponsoring Cancer Center

University of Pittsburgh Cancer Institute
Center for Pathology and Oncology Informatics

2. Workspace

Clinical Trials Workspace

3. Projects or Activities

– De-identification of both structured clinical data and narrative text

4. Workspace needs the project meets

– De-identification allows the data to be shared among participating institutions

5. Stage of project maturity

– 2 end-users at parent center; has de-identified over 200,000 documents for clinical research.

6. Technical details of Tools

Technical details of Tools

a. Software Architecture (These will likely be preliminary)

i. System design

De-ID runs in a Windows environment as a stand-alone PC application. It uses a set of heuristics to identify the presence of any of the HIPAA 17 specific identifiers within electronically stored medical text or tab-delimited text files. Supplemental dictionaries of geographic locations, hospital names, popular names found in the U.S. Census are also used to locate identifiable text. The UMLS Metathesaurus is utilized to ensure that words or phrases that are medical terms are preserved.

De-ID replaces identifiable text with specific tags. Names found multiple times in the report are consistently replaced with the same tag to improve readability of the report. The downside of applying De-ID is the removal of a small amount of clinical information during the de-identification process. In our work to date, we have found only minor problems with this. Most of the inappropriately de-identified text (overmarkings) consists of (1) addresses that contain commonly used words (e.g., the "MI" in Lansing, MI is confused as being an abbreviation for myocardial infarction), and (2) names that are medical terms but not in the UMLS (i.e. Hickman catheter). Over time, we have been identifying these problems and augmenting De-ID to address them.

ii. Component details

Reader

The De-ID application contains two layers. The first layer, named Reader, extracts the text from a text file or collection of text files. The text is read as a collection of Tokens. Each Token consists of one single word or number and the surrounding punctuation and/or whitespace

characters. The Reader code groups these tokens into sentences. For example, a possible sentence might be, "Mr. Jones is a 45-year-old male from Greentree."

Reader also attaches some information to each token in a bit vector. This information is used internally by Reader to identify sentence boundaries, but is also useful to De-ID.

DeIdentifier

As each token is read by the Reader, the second layer, the DeIdentifier (De-ID), looks up the token in its dictionaries. The dictionaries give information to De-ID about the type of word that token contains. The dictionaries are editable by users; making it possible to customize the dictionary to meet specific requirements of the user site (i.e. hospital names, sports teams (i.e. the person is the quarterback of the Steelers))

Identifiers are located in the text by the firing a set of rules a sentence at a time. The rules are fired in three passes over the text.

The first pass: During the first pass, phrases from the UMLS are identified. These phrases will not be used as candidates for de-identification during the later two passes.

The second pass: Most of the work is done during the second pass over the sentence. Each token in the sentence is considered from left to right. Depending on the bits that are set in the token certain rules will fire. The order that rules fire is important to avoid mistakes in de-identification. Once a rule successfully invoked for a given segment of text, no other rule will be allowed to be invoked for that same segment.

The third pass: On the final pass through the sentence, any missed community or person names should be discovered.

Linkage file

During de-identification, each record in the set is assigned a unique (sequential) record identifier. This identifier is stored in a linkage file and associated with the original header data. If more data needs to be found and de-identified later, this linkage file can be consulted to re-identify the patient of interest. Only a "trusted broker" should have access to the linkage file.

iii. Relevant standards

De-ID accepts documents in an XML formatted document and is able to produce the output in XML format. It utilizes the UMLS for defining medical terms and phrases.

- iv. UML schematics (if valid)
- v. Size of project installed software base

The software is installed in 2 locations at the University of Pittsburgh. It is available only to certified honest brokers within the University Health system. It is also run by a commercial company processing surgical pathology reports.

b. Development Environment (tools, languages, bug tracking, etc.)

De-ID is written in C++. It utilizes the CBMI help desk tool for reporting bugs.

7. Does the project make use of existing standards

De-ID can de-identify both structured and unstructured text, and exports de-identified text in either text or XML formats

8. Does other software in the community meet this need? Is this software open source? Can it be harnessed?

Software is not currently open source but may be available for licensing either through the University or through a third-party who is designated by the University as the licensing agent.

9. Points of possible interoperability with other caBIG systems

(This might include communication with other caBIG databases, use of caCORE APIs, caBIG-compatible APIs, etc.)

De-ID has an API so that it can be accessed within other applications. It does not currently integrate into other caCORE APIs or CDE.

10. What resources are proposed to achieve caBIG interoperability?

- a. Developmental requirements
 - i. Software (re)engineering
Current work on DE-ID API would need to be completed as well as revising the current heuristics to deal with formats (i.e. all upper case letters) that may be found in clinical documents not already encountered.
 - ii. Standards adoption
 - iii. Platform migration
- b. Infrastructure
 - i. Facilities
Office space for development team (n=3).
 - ii. Management tools
De-ID would need a variety of auditing tools to be developed to ensure compliance and accuracy. This might include the generation of a log file with number of items de-identified; types of de-identification tags, etc.
 - iii. Personnel
Number of personnel required would depend on the specifics of the licensing agreement, and whether an adopter wished to use the system as it stands now, or would require additional customizations. Determining those requirements would be possible only after auditing the compliance and accuracy of the tool in another organizational setting.

11. Draft 12-month work plan, with milestones to achieve caBIG interoperability.